

Lab 2 – Working with Data Types and Operators

Aims:

- To develop an understanding of the basic use of variables, arrays and expressions.
- To understand the basic implementation of form submission and passing variable between web pages.

Getting Started:

Create a new folder named "**lab2**" under the "C:\htdocs" folder on your computer. Save today's work in this folder.

Task 1: Using variables, arrays and operators

With a budget of \$50 daily, write the script to generate a short report which contains:

- the days which exceed the daily budget
- the daily average of expenses

The daily expenses are provided in the array below and index 0 is referring to Monday.

```
$expenses = array(30, 50, 25, 70, 48, 60, 90);
```

Step 1:

Create a file "**module1.php**" with a PHP script that declares and initialises an array named `$expenses` as shown above. Then write the scripts for the above scenario.

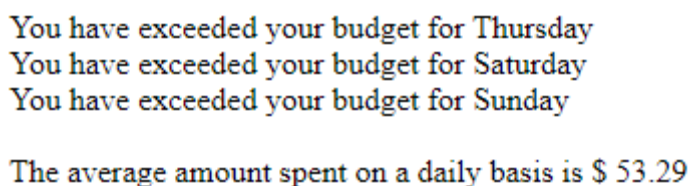
Set the title of the web page to Lab 2. In the web page content, display a heading that shows "Learning How to Use Arrays" using the second largest heading text.

The output statements is shown in step 2. Use `round()` function to format the average result to 2 decimal places.

Step 2:

Test in the browser, and check that the page is valid.

The following screenshot shows the sample output of "**module1.php**".



```
You have exceeded your budget for Thursday  
You have exceeded your budget for Saturday  
You have exceeded your budget for Sunday  
  
The average amount spent on a daily basis is $ 53.29
```

Task 2: Experimenting on arrays

Step 1:

Create a file "**daysarray.php**". Without using the `array()` construct, write a PHP script that declares and initializes an array named `$days` with the days of the week Sunday, Monday, etc.

Use output statements to display "The days of the week in English are:" along with the values in the `$days` array.

Step 2:

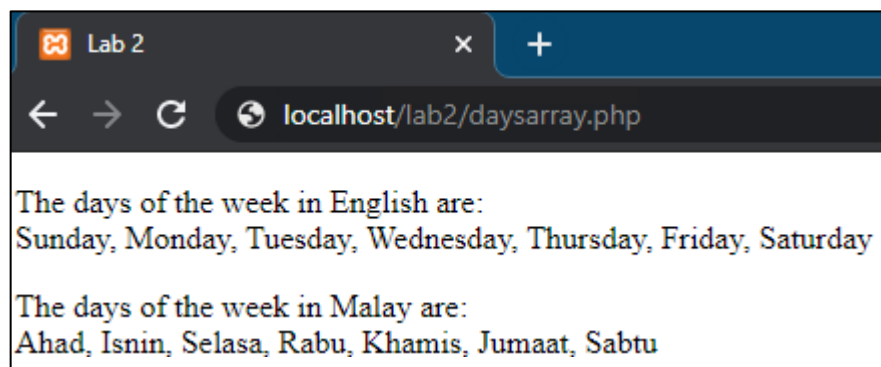
Test in the browser, and check that the page is valid.

Step 3:

Reassign the values in the `$days` array with the days of the week in Malay, Sunday is *Ahad*, Monday is *Isnin*, Tuesday is *Selasa*, Wednesday is *Rabu*, Thursday is *Khamis*, Friday is *Jumaat*, and Saturday is *Sabtu*.

Then use output statements to display "The days of the week in Malay are:" along with the Malay values in the `$days` array.

The following screenshot shows the sample output of "**daysarray.php**".



Task 3: Using expression and looking up built-in functions

Step 1:

Create another file "**iseven.php**" with a script that declares a variable `$num` with a numeric value. To handle the possible case of having a floating-point numeric number, you will need to add the `round()` function to convert the initial value to the nearest whole number.

Use the `?:` conditional operator to determine whether the number is even. Then, use an output statement to display a message to show the results.

Step 2:

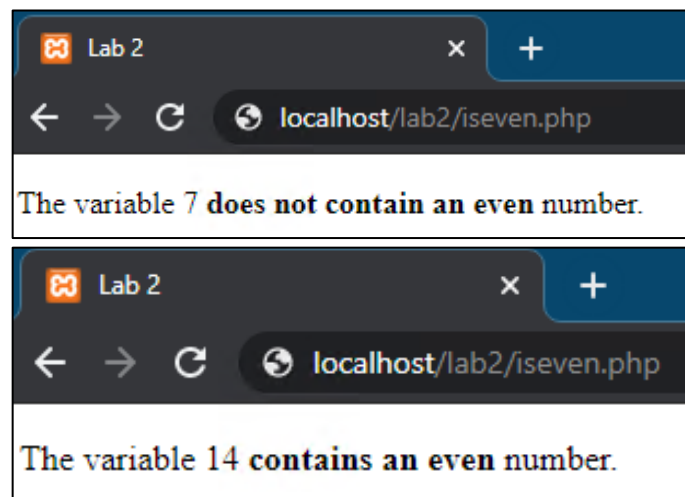
Test the script by modifying the variable value, re-saving the script to the server each time, and refreshing the page in your browser.

Function Description and Examples

`round()`

<http://php.net/manual/en/function.round.php>

The following screenshot shows the sample output of "**iseven.php**".



Task 4: Working with form

Create a form in another page "**num_input.php**" that passes the variable to "**iseven.php**" (from Task 3) using the GET method. Add code in "**iseven.php**" to receive the variable.

Hint: Use `$_GET[...]`.

See *Predefined Variables*, Superglobals and examples: <http://php.net/manual/en/reserved.variables.php>

The following screenshot shows the interface of "**num_input.php**".

